H-038

UI パーツを対象とした one-shot 画像分類手法の提案
One-Shot Image Classification method for UI Parts

四十崎 航† 篠澤 佳久†† 飯村 結香子‡ 斎藤 忍‡
Wataru Aizaki Yoshihisa Shinozawa Yukako Iimura Shinobu Saito

1. はじめに

画像認識や自然言語処理などの多様なタスクにおいては，深層学習による研究によって急速な発展を遂げている．しかしそうした手法の多くは教師あり学習をもとにしており，タスクに応じた大量の学習データを必要とする．一方，タスクによっては十分な学習データが得られないことやデータ収集のコストを許容できないことがある．そのため，近年は大規模データセットにより事前学習されたモデルや自己教師あり学習モデルを用いることで少量のデータから特徴表現を獲得することが試みられている[1][2].

画像認識のタスクの一つとして，GUI 画像における基本的な要素（UI パーツ）を検出するモデルの作成が試みられてきた[3]．GUI 画像から再利用可能な UI パーツを抽出できるモデルを構築することによって，開発コストの削減が期待できるためである．

本研究では，一般的な深層学習の物体検出タスクと同様に，第一に UI パーツの位置を検出し，第二にクラス分類を行う二段階のモデルを構築する．位置の検出については複数の手法による研究がされており[4]，特に本研究においては，クラス分類の手法について検討する．ただし GUI 画像に含まれる UI パーツの種類は多様であり，かつそれぞれのデータ数も少ないために大量の学習データを収集することは難しい．そのため，教師あり学習を行わずにクラス分類を行う手法を提案する．

2. 提案

本研究では，UI パーツのクラス分類を行う手法を提案する．これは UI パーツを検出するタスクを，位置を検出するタスクとクラスを特定するタスクに分けたとき，後者に相当する．さらに，クラス分類は one-shot で行う．つまり，UI パーツの各クラス中の 1 つを正解ラベルが付けられたサポートセット，他の正解ラベルの無いデータをクエリ—セットとする．

研究対象である GUI 画像や，それから検出される UI パーツは画像データとして扱うことができる．そこで画像特徴を抽出する．また GUI 画像には文字列が含まれており，UI パーツによって異なっている．そこで文字列を考慮した分類を行うため，文字特徴を抽出する．特徴抽出後，各クラスのサポートセットとクエリ—セットの画像特徴と文字特徴を考慮した類似度を計算し，クラス分類を行う．

3. 提案手法

ラベル付きデータが少数である場合，少数のデータのみから高性能なモデルの作成は困難である．そこで本研究においては，下記の 3 種類の事前学習済みモデルを用いて UI パーツから特徴抽出を行う．

手法①：ResNet による推論
手法②：SimCLR による推論
手法③：BERT による推論

3.1 ResNet による推論（手法①）

ある画像分類タスクにおいて学習データが十分に得られない時，ImageNet などの大規模データセットにより事前学習されたモデルを転用する転移学習が有効である．これは，画像を学習対象とした場合，そのようなモデルの下位層では普遍的な特徴が抽出できるためである．本研究では転移学習はせず，普遍的な特徴を抽出する特徴抽出器として，ImageNet で学習された ResNet-18[5] の畳み込み層からの出力を平坦化して 25,088 次元の特徴ベクトルを抽出する．

3.2 SimCLR による推論（手法②）

SimCLR[1]は自己教師あり学習の一手法であり，対照学習によって画像の特徴表現を得ることができる．対照学習は画像のペアのそれぞれ意味するものが近い場合は類似度を大きく，一方で遠い場合には類似度を小さくなるように，データを正解ラベルなしで学習する手法である．手法①と同様に，特徴抽出器として ResNet-18 の畳み込み層からの出力を平坦化して 25,088 次元の特徴ベクトルを抽出する．

3.3 BERT による推論（手法③）

BERT[2]は 2 種類の言語タスクを大規模データセットによって事前学習した深層学習モデルである．手法③の手順を以下に示す．

(1) Tesseract[6]を用いて対象の UI パーツ画像に含まれる文字列の抽出を行う．
(2) 抽出された文字列をすべて半角スペースで結合し，一文とみなす．本研究で用いるデータは大半の文字列が英語であるため，結合した文を英文とみなす．図 1 に UI パーツ画像から作成した英文の例を示す．

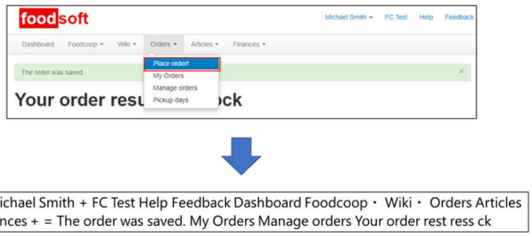

図 1　英文作成の例

(3) 英文を BERT の事前学習済みモデルにより推論を行い，出力値を得る．出力のうち，「'[CLS]'」トークンに対応する 768 次元の特徴ベクトルを抽出する．

†慶應義塾大学大学院 理工学研究科　Graduate School of Science and Technology, Keio University
††慶應義塾大学 理工学部　Faculty of Science and Technology, Keio University
‡日本電信電話株式会社 コンピュータ&データサイエンス研究所　Nippon Telegraph and Telephone Corporation

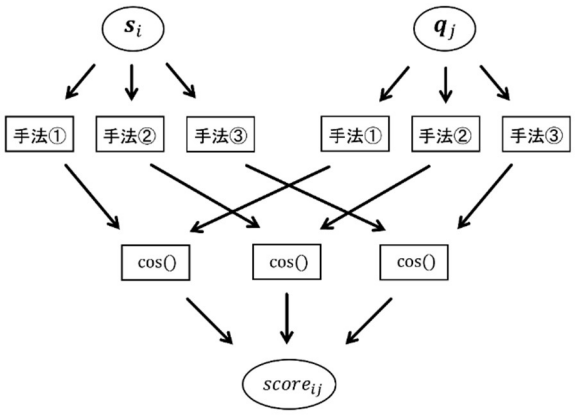

図 2　提案手法の流れ

提案手法の概要を図 2 に示す．上記の 3 種類の手法をそれぞれすべての UI パーツに適用して特徴ベクトルを得る．そしてサポートセットとクエリ―セットの 1 つずつをペアのデータとして，それぞれ類似度を計算する．類似度はコサイン類似度により求める．つまり，$\boldsymbol{v}_1$と$\boldsymbol{v}_2$の類似度は式(1)で定義される．

$$score = \frac{\boldsymbol{v}_1 \cdot \boldsymbol{v}_2}{\|\boldsymbol{v}_1\|\|\boldsymbol{v}_2\|} \quad (1)$$

図 2 の「cos()」はこの計算処理を表す．よって，サポートセットのi番目のデータ$\boldsymbol{s}_i$とクエリ―セットのj番目のデータ$\boldsymbol{q}_j$の類似度は式(2)で表される．

$$score_{ij} = \sum_{m=1}^{3} \lambda_m \times \frac{f_m(\boldsymbol{s}_i) \cdot f_m(\boldsymbol{q}_j)}{\|f_m(\boldsymbol{s}_i)\|\|f_m(\boldsymbol{q}_j)\|} \quad (2)$$

ただし，mの値は上記の手法①②③に対応し，f_mはこれらの手法による特徴抽出処理を指す．また，λ_mは手法ごとの重みを表すハイパーパラメータである．ある$\boldsymbol{q}_j$に対して，$score_{ij}$の値が最大となるiを求める．サポートセットは各クラス 1 つのみであるため，iはクラスのラベルに対応しており，そのクラスに分類する．

4. 評価実験

本研究では，評価実験のためにオープンソースソフトウェア（OSS）の Web システム上にて 381 枚の Web ページ画像を収集した．そして画像に含まれるヘッダーやフッター，メッセージなど業務的に意味をもつ情報のまとまりを UI パーツと定義した．手動で UI パーツを検出した後，データ数が 11 以上のクラスから 11 枚の UI パーツ画像を選択し，これを評価実験の対象とした．つまり，分析対象のデータは 30 クラス，330 枚の UI パーツ画像である．

評価実験の手順を以下に示す．

(1) それぞれのクラスからサポートセットとする UI パーツ画像を 1 枚選択して，30 枚をサポートセット，残りの 300 枚をクエリ―セットとする．ただし，一度サポートセットとして選択した UI パーツ画像は非復元抽出により再びサポートセットとして選択しない．

(2) クエリ―セットのデータをそれぞれクラス分類する．

(3) 正解率などの評価指標を求める．

(4) (1)に戻る．(1)から(3)を 11 回繰り返し，すべてのデータが一度サポートセットとして選択された後，(5)に進む．

(5) (3)を 11 回繰り返して求めた評価指標の値の平均値を評価実験の結果とする．

表 1 に各手法によりクラス分類を行った結果を示す．提案手法のモデルは$\lambda_1 = 1, \lambda_2 = 0.2, \lambda_3 = 2$とした．また，正解率以外の指標についてはマクロ平均を示す．

表 1　評価実験の結果

モデル	正解率	適合率	再現率	F 値
手法①	0.777	0.812	0.777	0.766
手法②	0.753	0.804	0.753	0.740
手法③	0.669	0.722	0.669	0.654
提案手法	0.818	0.849	0.818	0.811

図 3 に，$\lambda_1 = 1$に固定してλ_2とλ_3を変化させたときの正解率の変化を示す．各行はλ_2の変化を示し，行名のアンダーバー以降の値がハイパーパラメータの値である．同様に各列はλ_3の変化を示す．

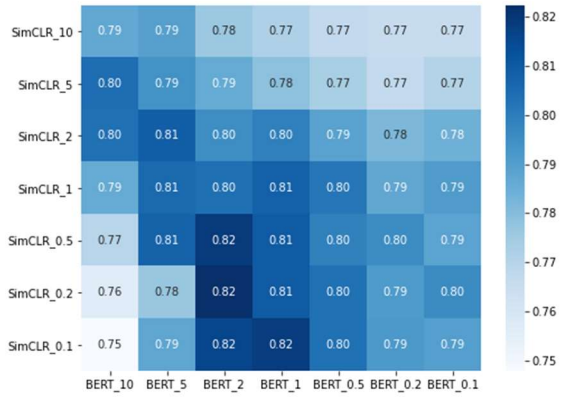

図 3　ハイパーパラメータと正解率の関係

表 1 より，提案手法によってすべての評価指標において他の手法よりも高い精度が得られた．よって，3 種類の手法を組み合わせることで UI パーツの分類に適切な類似度が求められることを示した．また，図 3 から適切なハイパーパラメータの設定を行う必要があることが分かった．

5. まとめ

GUI 画像に含まれる UI パーツに対して，複数の特徴抽出器を用いて各クラス 1 つのデータのみから分類を行った．画像特徴だけでなく，文字列を抽出して文字特徴を用いることで，単体モデルよりも高性能な分類モデルとなることを示した．今後は GUI 画像から UI パーツの候補領域を検出するモデルと組み合わせ，汎用的な UI パーツ検出モデルの構築に取り組む予定である．

参考文献

[1] Ting Chen *et al.*, A Simple Framework for Contrastive Learning of Visual Representations, In ICML, pp. 1597-1607, 2020.
[2] Jacob Devlin *et al.*, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, In NAACL-HLT, pp. 4171-4186, 2019.
[3] Toby Jia-Jun Li *et al.*, Screen2Vec: Semantic Embedding of GUI Screens and GUI Components, In CHI, pp.1-15, 2021
[4] 加藤聡太郎ほか，Web ページ画像からの UI パーツの抽出，情報処理学会，第 84 回全国大会講演論文集，Vol.2，6S-04，pp.445-446, 2022.
[5] Kaiming He *et al.*, Deep Residual Learning for Image Recognition, In CVPR, pp. 770-778, 2016.
[6] https://github.com/tesseract-ocr/tesseract（2022 年 6 月参照）